

**ASSIGNMENT SCHEDULING CAPABILITY FOR UNMANNED AERIAL VEHICLES –
A DISCRETE EVENT SIMULATION WITH OPTIMIZATION IN THE LOOP APPROACH TO SOLVING A
SCHEDULING PROBLEM**

Darryl K. Ahner

Training and Doctrine Command
Analysis Center
Monterey, CA 93943, U.S.A.

Arnold H. Buss

MOVES Institute
Naval Postgraduate School
Monterey, CA 93943, U.S.A.

John Ruck

Rolands & Associates Corp.
500 Sloat Ave
Monterey, CA 93940, U.S.A.

ABSTRACT

Many military planning problems are difficult to solve using pure mathematical programming techniques. One such problem is scheduling unmanned aerial vehicles (UAVs) in military operations subject to dynamic movement and control constraints. This problem is instead formulated as a dynamic programming problem whose approximate solution is obtained via the Assignment Scheduling Capability for UAVs (ASC-U) model using concepts from both simulation and optimization. Optimization is very effective at identifying the best decision for static problems, but is weaker in identifying the best decision in dynamic systems. Simulation is very effective in modeling and capturing dynamic effects, but is weak in optimizing from alternatives. ASC-U exploits the relative strengths of both methodologies by periodically re-optimizing UAV assignments and then having the simulation transition the states according to state dynamics. ASC-U thus exploits the strengths of simulation and optimization to construct good, timely solutions that neither optimization nor simulation could achieve alone.

1 INTRODUCTION

Military operations are dynamic, complex series of events. The problems associated with military operations are naturally also dynamic and complex. This paper describes the dynamic problem associated with the allocation of UAVs from their owning military units to mission areas over time in Section 2. The complex dynamics of the problem are evident from ground control stations and the launch and recovery sites, from which the UAVs are launched, return, and are recovered, each of which move from one timestamped waypoint to the next throughout the scenario. The use of fuel during flight and the recovery time once at a launch and recovery site are sources of additional dynamics.

In Section 3, the paper describes why classical mathematical programming or simulation alone have difficulties solving this highly dynamical scheduling problem. The strengths and weaknesses of the two approaches are discussed. The Dynamic Allocation of Fires and Sensors simulation, on which ASC-U is built, is described in Section 4.

A dynamic programming formulation is explained in Section 5. This section explains how a dynamic programming approximation can exploit the special structure of the problem to obtain quality solutions quickly. Then, a solution technique that makes use of the strengths of a combined simulation and optimization approach is described in Section 6, and, finally, initial results, insights, and conclusions are discussed in Section 7.

2 THE UNMANNED AERIAL VEHICLE SCHEDULING PROBLEM

We consider a dynamic unmanned aerial vehicle routing problem. In addition to unmanned aerial vehicles (UAVs), this problem accounts for moving launch and recovery sites (LRSs) from which UAVs are launched and must return, moving ground control stations (GCSs) from which UAVs must be controlled at every instant during flight and that can control a finite number of UAVs at any one time, and stationary missions from which a time-dependent value is received if a UAV with the appropriate sensor visits. The objective is to determine the maximum sum of mission values that can be accomplished given the number of each type of UAVs and their performance characteristics of flight time and speed.

Missions are characterized by a set of sensor requirements at a Cartesian coordinate. Each sensor requirement has a start and end time, a type of sensor required and a value rate. When a UAV with the correct sensor arrives at the same location as a sensor requirement, and that requirement does not yet have a sensor accomplishing that requirement, the UAV receives credit for accomplishing

Report Documentation Page		Form Approved OMB No. 0704-0188		
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.				
1. REPORT DATE DEC 2006	2. REPORT TYPE	3. DATES COVERED 00-00-2006 to 00-00-2006		
4. TITLE AND SUBTITLE Assignment Scheduling Capability for Unmanned Aerial Vehicles - A Discrete Event Simulation with Optimization in the Loop Approach to Solving a Scheduling Problem		5a. CONTRACT NUMBER		
		5b. GRANT NUMBER		
		5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S)		5d. PROJECT NUMBER		
		5e. TASK NUMBER		
		5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School,MOVES Institute,Monterey,CA,93943		8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)		
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited				
13. SUPPLEMENTARY NOTES Proceedings of the 2006 Winter Simulation Conference, 3-6 Dec, Monterey, CA				
14. ABSTRACT see report				
15. SUBJECT TERMS				
16. SECURITY CLASSIFICATION OF:		17. LIMITATION OF ABSTRACT Same as Report (SAR)	18. NUMBER OF PAGES 8	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified		

that sensor requirement. The value received is the UAV's time-on-station multiplied by the value rate of the sensor requirement that is satisfied. The UAV receives value until the UAV departs or the sensor requirement ends. Since UAVs carry sensor packages with potentially many sensors, multiple missions may be satisfied and these values are additive.

UAVs are characterized by a unique name, a type, a starting and ending LRS, an operating time, a speed, a transition time for recovery and refueling, an allowable operating radius from a controlling GCS, a sensor package, and an available start time and an end time. The unique name identifies each UAV in order to capture which sensor requirements it accomplished and its state throughout the problem. The type of UAV is important to ensure that only GCSs and LRSs that are compatible with a given UAV are used for control and recovery, respectively. Starting and ending LRS designations for each UAV allow for the initial and terminal conditions of the problem and support the requirement that a UAV should begin and end at the appropriate organizational unit. Operating time and speed dictate whether a mission is feasible for a given UAV and how much time-on-station the UAV can achieve before it must return to an LRS for recovery and refueling. The transition time for recovery and refueling provides the time a given UAV must spend at an LRS before it can be launched again. The allowable operating radius is the maximum distance a UAV can be from a GCS and still be controlled by that GCS. A sensor package is a set of sensors that can be assigned to a UAV for a particular flight. The available start and end times for a UAV determine the times it should be considered as available to fly.

Sensor packages are sets of sensors, weapons, or communications capabilities. A sensor package is assigned to a UAV and determines what sensor requirements that the UAV can accomplish. Value is only gained for a sensor requirement if the UAV's sensor package has the appropriate sensor in its set.

LRSs are characterized by a unique name, a UAV type, a Cartesian coordinate, and a capacity. The LRS is where the UAV must originate and end its flight. The LRS is also where the UAV will, upon landing, transition through recovery and refueling before becoming available to depart to accomplish more missions. LRSs may move over time.

GCSs are also characterized by a unique name, a UAV type, a Cartesian coordinate, and a capacity. The GCS controls the UAV during flight and may provide the communications link over which its information is passed when necessary. GCSs may move over time.

3 MATHEMATICAL PROGRAMMING VS. SIMULATION

In a mathematical programming problem, one seeks to minimize or maximize a real function of real or integer variables, subject to constraints on the variables. An integer programming problem takes the form:

$$\begin{aligned} & \max c^T x \\ \text{subject to } & Ax = b, \text{ where } x \in Z \end{aligned}$$

where c is the vector of costs or rewards, x is the decision variable, and Z is the set of integers. Using this formulation to model the UAV scheduling problem has several pitfalls. If uniform time steps are used for travel to and from the missions the solution to an integer program may not be feasible. Additionally, the dynamics involved are difficult to model in the constraint function, $Ax = b$.

Even with effective modeling, the dynamics and time-steps of the problem cause the number of decision variables to suffer from high dimensionality so that extensive decomposition techniques may be needed to solve the problem to optimality or to even find a near-optimal feasible solution. Solution times for large integer problems are notoriously long.

Mathematical programming is a powerful tool for more static problems. However, for dynamic problems it suffers from an exponential number of variables and considerable solution challenges.

On the other hand, simulation allows straightforward and efficient modeling of dynamics, especially of moving entities (Buss and Sanchez 2005). Although simulation allows movement, representation of state changes and, easy collection of statistical measures of, by itself it does not provide the decision capacity needed to solve effectively tell UAVs where to go next. Myopic decision policies are often used within a simulation but these policies would be inadequate for the UAV scheduling problem which requires sequential decisions over time and space in a GCS capacity constrained environment.

The need for complex decisions prohibits the use of a traditional simulation to determine the effects of a given allocation of UAV systems. The complexities of the problem caused by moving LRSs and GCSs and their capacities prohibit the problem from being easily formulated using mathematical programming formulations which are better suited for stationary problems. Instead, the problem is viewed and formulated as a dynamic programming problem and, furthermore, solved using approximate dynamic programming techniques using a combination of simulation, to model movement, and simple mathematical programming, to model the allocation decisions.

4 DYNAMIC ALLOCATION OF FIRES AND SENSORS SIMULATION

The Dynamic Allocation of Fires and Sensors (DAFS) simulation is a low-resolution, constructive entity-level simulation framework designed for combat modeling (Havens 2002; Buss, Ahner, and Ruck 2006). The DAFS framework consists of a Discrete Event Simulation Model with embedded optimization, Extensible Markup Language (XML) input and output modules, and an output analysis package (Ahner 2005). Through the use of an interchangeable component-based architecture, the simulation provides the user extensive ability to modify entities, configurations, simulation parameters and data output. DAFS, using a low resolution approach, runs fast and is easy to set up.

DAFS has a component-based architecture. Within the simulation model, there are two types of components that work together to give DAFS its overall capability, physical components and functional components. Physical components represent physical items such as sensors and munitions. Functional components control some action within the simulation. One functional component is the constrained value optimizer (CVO).

The Constrained Value Optimizer (CVO) is the component that provides a myopic optimal solution for the current state of the simulation. The CVO allows for the reallocation of BLUE platform resources to improve the BLUE forces near-term chances of successfully completing its mission.

The CVO does not perform global dynamic optimization which would be computationally intractable for large problems. Instead, the CVO takes the current state of BLUE resources, compares the capabilities of BLUE forces to the state of demands, and through its mathematical formulation of the problem determines the ‘optimal’ actions to take according to the current states modeled in the mathematical programming formulation. Since this approach has a control theory flavor to it, it has been called model predictive control. The CVO can employ either a mathematical programming solver or a heuristic.

As currently implemented, the objective function formulated for mathematical programs is solved using LP_Solve 5.5 (Lp_Solve 2005). LP_Solve 5.5 is an implementation of the simplex method for linear program formulations and integer program formulations using branch and bound. LP_Solve is implemented in the C programming language, so DAFS communicates with it using the Java Native Interface (JNI).

In DAFS, the constrained value optimizer is implemented as an interface. A CVO is constructed and is called periodically to cue UAV allocation decisions given the current state of the simulation.

This use of a control, given a current state, falls within the field of dynamic programming, a methodology for op-

timization of sequential decisions. The Dynamic Programming formulation of the UAV assignment problem is discussed in the next section.

5 DYNAMIC PROGRAMMING FORMULATION

Consider a finite state and discrete time horizon dynamic programming problem. Let S be the state space of the system. The finite time horizon is $t = 0, \dots, T$ where the time steps may or may not be uniform. The state $S_t \in S$ represents the state at time $t = 0, \dots, T$. A decision vector \bar{u}_t that acts on the system is selected from a finite set U at each time step. The state evolves according to a state equation which has the form

$$S_{t+\tau} = f_1(S_t, \bar{u}_t, \tau)$$

where f_1 is a function describing the systems dynamics, S_t includes the state of the missions, LRSs, GCSs, and UAVs which are defined in the next section, and τ is the time to the next decision event.

A policy is a mapping $\Pi(S_t): S_t \rightarrow U$ that determines a decision as a function of the state, i.e. $\bar{u}_t = \Pi(S_t)$. We represent the one period contribution to the reward as $C_t(S_t, \bar{u}_t)$. We express the T -stage value to be maximized as the value of the summation of the T costs:

$$\max_{\bar{u}_t \in \Pi(S_t)} \sum_{t=0}^T C_t(S_t, \bar{u}_t)$$

The cost-to-go or future value function at time t is

$$J_t(S_t) = \max_{\bar{u}_t, \dots, \bar{u}_T} \sum_{\tau=t}^T C_\tau(S_\tau, \bar{u}_\tau) \\ s.t. S_{t+\tau} = f_1(S_t, \bar{u}_t, \tau) \text{ for } t = 0, \dots, T-1$$

We can solve for $J_t(S_t)$ using the recursive equation:

$$J_t(S_t) = \max_{\bar{u}_t \in U} C_t(S_t, \bar{u}_t) + J_{t+\tau}(S_{t+\tau}) \text{ for } t = 0, \dots, T-1$$

It is in this context of dynamic programming that we describe our formulation and methodology.

5.1 State Space Description

The state $S_t \in S$ is defined as

$$S_t = \left\{ \bigcup_{i \in UAV} S_t^{UAV_i}, \bigcup_{j \in GCS} S_t^{GCS_j}, \bigcup_{k \in LRS} S_t^{LRS_k}, \bigcup_{l \in Mission} S_t^{Mission_l} \right\}$$

where UAV , GCS , LRS , and $Mission$ represent the sets of UAVs, GCSs, LRSs, and missions, respectively. The state of these individual components represent an integral part of the overall state of the system.

The UAV state, $S_t^{UAV_i}$, is defined as

$$S_t^{UAV_i} = \left\{ \begin{array}{l} \langle x, y \rangle_t^{UAV_i} \\ t_{flight_time,t}^{UAV_i} \\ t_{recovery_time,t}^{UAV_i} \\ Airborne_State_t^{UAV_i} = \{atLRS, airborne\} \\ sensor_package_t^{UAV_i} \\ UAV_type^{UAV_i} \end{array} \right\}$$

The set accounts for the Cartesian position, $\langle x, y \rangle$, of each UAV at time t , the flight time remaining (if applicable, \emptyset otherwise), the recovery time remaining if located at an LRS (if applicable, \emptyset otherwise), whether the UAV is located at an LRS and not airborne (*atLRS*) or in flight (*airborn*), the sensors that the UAV is carrying at time t , and the type of UAV including all the performance characteristics associated with that UAV type. Performance characteristics for a UAV include

- Speed,
- Operating time,
- Transition time,
- Sensor package,
- Available start time, and
- Operating radius.

The GCS state, $S_t^{GCS_j}$, is defined as

$$S_t^{GCS_j} = \left\{ \begin{array}{l} \langle x, y \rangle_t^{GCS_j} \\ capacity_remaining_t^{GCS_j} \\ UAV_type^{GCS_j} \end{array} \right\}$$

The set accounts for the Cartesian position, $\langle x, y \rangle$, of each GCS at time t , the capacity remaining, and the type(s) of UAV(s) that the GCS can control. The maximum radius of control for a GCS is UAV dependent and is given in the set $S_t^{UAV_i}$ as a UAV performance characteristic.

The LRS state, $S_t^{LRS_k}$, is defined as

$$S_t^{LRS_k} = \left\{ \begin{array}{l} \langle x, y \rangle_t^{LRS_k} \\ capacity_remaining_t^{LRS_k} \\ UAV_type^{LRS_k} \end{array} \right\}$$

The set accounts for the Cartesian position, $\langle x, y \rangle$, of each LRS at time t , the capacity remaining, and the type(s) of UAV(s) that the LRS can land and recover. The recovery

time for a GCS is UAV dependent and is given in the set $S_t^{UAV_i}$ as a UAV performance characteristic.

The mission state, $S_t^{Mission_l}$, is defined as

$$S_t^{Mission_l} = \left\{ \bigcup_{s \in Mission_l} \langle x, y \rangle^{Mission_l} \right\}$$

where a sensor requirement, $sensor_requirements_s$, is defined as:

$$sensor_requirements_s = \left\{ \bigcup_{t_1 \in Time_segments_s} (open_time_available_{t_1}, end_time_available_{t_1}) \right\}$$

The set accounts for the Cartesian position, $\langle x, y \rangle$, of each mission which is constant and the sensor requirements of the mission to include sensor type, open time segments, and the value rate received during the open time segments. The value rate is multiplied by the UAV time-on-station if the UAV has the appropriate sensor in its sensor package.

5.2 Control Space

The dynamic programming formulation can be solved by solving the recursive equations:

$$J_t(S_t) = \max_{\bar{u}_t \in U} C_t(S_t, \bar{u}_t) + J_{t+\tau}(S_{t+\tau}) \text{ for } t = 0, \dots, T - \tau_f$$

where τ_f is the time difference from the end of the time horizon and the last applied control, $\bar{u}_{t-\tau_f}$. Since all missions are known *a priori*, ideally the optimal control, \bar{u}_t , could be obtained by

$$\bar{u}_t = \arg \max_{u_t \in U} C_t(S_t, u_t) + J_{t+\tau}(S_{t+\tau}) \text{ for } t = 0, \dots, T - \tau_f$$

Unfortunately, because of the combinatorial nature of the problem, this formulation suffers from the curse of dimensionality in both the state and control space. Therefore, the cost-to-go function, $J_t(S_t)$, is not easily obtainable. The constrained value optimizer of DAFS can be used instead to obtain an approximate solution using the mission state “in the near future.” This value “in the near future” is substituted for $J_t(S_t)$. This mission state will be described further in Section 6.

5.3 State Dynamics

DAFS implements a discrete event world view using event graph methodology developed originally by Schruben (1983). Event graph methodology assumes an event list

which advance to the next event on the list and can be represented graphically, as shown in Figure 1. Each node represents a state transition and each arc represents a scheduling relationship through which an event can cause other events to be placed on the event list. In the example shown, a mission could be assigned to a UAV platform by the CVO object and that mission might be out of the range of all sensors on the platform. This event triggers a Start Move event for the platform in order to move the platform within the range for the sensor. When the Start Move event occurs, the End Move event would be placed on the event list with a time delay of t , which in this case would be the amount of time it would take the platform to travel to its destination

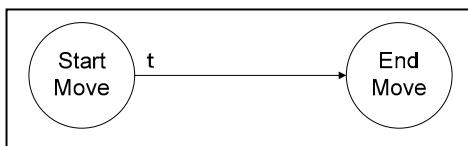


Figure 1: Event Graph Example

DAFS' event graph methodology handles all of the dynamic programming state dynamics in an efficient way. The simulation's can easily represent the dynamics, especially of entities moving and changing Ground Control Station capacities. For further discussion of modeling movement using Event Graph Methodology, see Buss and Sanchez (2005).

6 THE ASSIGNMENT SCHEDULING CAPABILITY FOR UAV SYSTEMS APPROACH

As mentioned previously, ASCU does not use simulation in the traditional manner. Instead, it uses simulation to transition and capture the state of the system dynamics, especially the movement of key entities (UAVs, GCSs, and LRSs) while applying controls using optimization in order to construct feasible UAV schedules.

ASCU periodically provides controls to available UAVs by solving an assignment problem based on a decision time window into the future. There are thus two parameters of the optimization logic of ASC-U: the duration of the decision time window (the amount of future time considered in each decision) and the time between re-optimizations. These parameters can be independently set in the input files.

When an optimization event occurs, ASC-U assigns currently available UAVs to Missions that are open within a fixed time horizon in the future that is dependent on the type of UAV being assigned. This assignment is made to maximize the total value obtained by assigning the available UAVs to Mission Areas that will have some sensor demand in the time horizon. This fixed time horizon is the

same for UAVs of the same type but differs for UAVs of differing types. The reason for this difference is that ASC-U considers each UAV type's maximum cycle time. A UAV's maximum cycle time is its maximum endurance time plus recovery time. This maximum cycle time is multiplied by a constant which is determined experimentally.

For example, consider 2 UAVs assigned to 4 missions as depicted in Figure 2. ASC-U performs optimal assignments at predetermined time intervals (1). At time t_0 , assume two UAVs are available, UAV1 and UAV 2. The tool considers all available UAVs with available GCS control and all missions with value in the fixed time horizon time window (missions 1,2,3 but not 4 for time t_0) (2). A UAV is assigned to a mission and can service any sensor requirement associated with that mission if it has the correct sensor. Assume the optimal assignment is UAV 1 assigned to Mission 1 and UAV 2 assigned to Mission 2. UAV 1 is launched immediately to arrive at Mission 1 as soon as possible (3). UAV 2 does not launch but is scheduled for launch to arrive just-in-time for the beginning of Mission 2. At time t_1 , UAVs that are scheduled to launch but have not yet launched are "unassigned" and considered for assignment (UAV 2 is unassigned) (2). UAV 2 is again assigned to Mission 2 and scheduled for launch to arrive just-in-time. UAV assignments do not change once they have been launched. By advancing in time, each UAV has a schedule constructed that reflects the optimal assignment for available UAVs considering the fixed time horizon in the future.

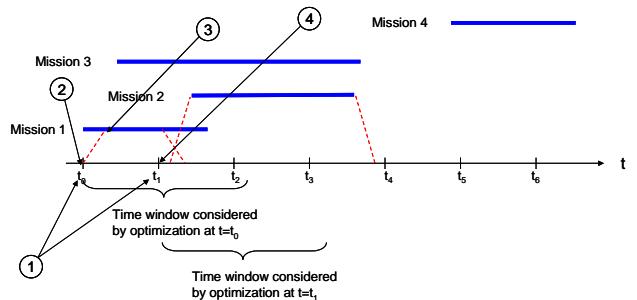


Figure 2: Assignment Dynamics Example

The value of each mission is determined by the user-defined valuation rate of sensor requirements for each mission. These user-defined sensor valuation rates can depend on the echelon requiring the mission or the priority given to the mission or a combination. The UAV receives a total value for missions by determining the time-on-station of the UAV and multiplying it by the value rate if the sensor requirement's end time is not exceeded or by the value rate from the UAV arrival to the sensor requirement's end time if the mission's end time is exceeded by the time-on-station of the UAV. These values are then added to determine the possible value of each mission if the UAV were

assigned to the mission. Finally, if the mission is not within a GCS footprint with capacity then the value is set to zero.

Figure 3 shows the different stages of ASC-U as it constructs the UAV schedules. Given a state of the system in the lower left of Figure 3, a periodically occurring optimization event is scheduled to occur. This optimization event first uses a Value of Potential Assignment Generator to calculate the value received within the UAV-type time windows for every possible assignment for every UAV. This value takes into account movement of entities and GCS range constraints. These assignment values along with constraints on available GCS capacity are handed off to the optimization. The optimization determines the maximum value of this optimization problem and hands these assignments to the UAV Scheduler. The UAV Scheduler determines when the UAV is launched so that the UAV arrives just-in-time if the mission start time minus the travel time to the mission is greater than the current time. The next optimization event is scheduled to take place after a delay of the optimization time interval. The Dynamic Simulation State Transitions then occur until that event occurs, so that it will generally be considering a very different state than the previous optimization.

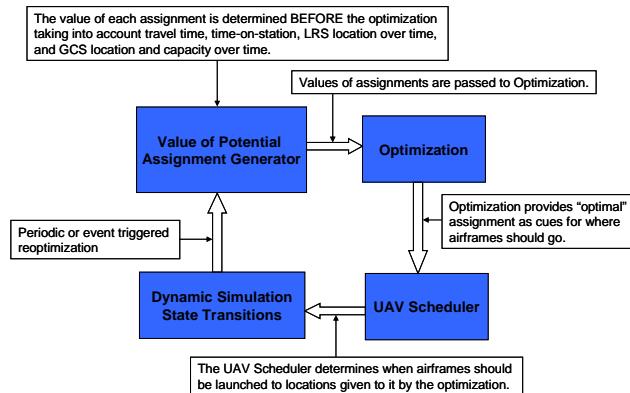


Figure 3: ASC-U Dynamic Cueing and Transitions

The following optimization problem is solved at each optimization event.

- Let A be the set of all mission areas with at least one mission active within the global time horizon.
- Let L be the set of all active LRS's
- Let G be the set of all GCS's.
- Let G_L the set of all GCS's assigned to LRS L .
- Let C_g the number of UAV's GCS g is capable of controlling.
- Let I_l be the sub-set of all UAV's at LRS $l \in L$ determined as follows: For each LRS $l \in L$, get n UAV's where n is the minimum of the number of ready UAV's, the LRS launch limit, the number of UAV's airborne, the sum over G_L of C_g , and the number of UAV's assigned to the GCS.

- Let J_l be the sub-set of all sensor packages currently located at LRS $l \in L$.
- Let $Y_{ga} = 1$ if mission area a is assigned to GCS g , 0 otherwise (by a heuristic discussed below).
- Let c_{ja} = The reward for a UAV with sensor package j being assigned to mission area a from the soonest possible arrival time of the UAV at the area to the end of the time horizon, $t + \Delta t^i$, for UAV i .
- $X_{ja} = 1$ if a UAV with sensor package j is assigned to mission area a , and 0 otherwise.

Then the formulation is:

$$\max \sum_{ja} c_{ja} X_{ja} \quad (1)$$

such that

$$\sum_j X_{ja} \leq 1 \forall a \in A \quad (2)$$

$$\sum_a X_{ja} \leq 1 \forall i \in I \quad (3)$$

$$\sum_{ga} Y_{ga} X_{ja} \leq C_g \forall g \in G \quad (4)$$

$$\sum_{j \in J_l a} X_{ja} \leq |I_l| \forall l \in L \quad (5)$$

- Maximize the value of Mission Areas covered.
- Assign only 1 UAV per Mission Area.
- Assign only 1 Mission Area per UAV.
- Do not exceed the GCS control limit.
- The number of sensors assigned cannot exceed the number of UAVs available to carry them.

Heuristic for determining Y_{ga} (assignment of Mission Areas' to GCS):

For each LRS and Mission Area a :

Let N_a the number of GCS's g that are in range of a UAV assigned to Mission Area a

For each Mission Area a , sorted by N_a :

For each GCS g , sorted by $\sum_a Y_{ga}$ (so far)

If a is in range of g , let $Y_{ga} = 1$.

Detail of determination of c_{ia}

For each UAV i

For each Mission Area a

Let t_o = the first time after the earliest arrival time that UAV i can gain value by being assigned to Mission Area a .

Let $t_l = \min(\text{the latest time UAV } i \text{ can remain at Mission Area } a, \text{ the end of the time horizon, } t + \Delta t^i, \text{ for UAV } i)$.

Let $K_a = \text{the set of all Missions located at Mission Area } a$.

Let $V_{i,k,t0,t1}$ = the value that UAV i gains from Mission k by being at Mission Area a .

Then, $c_{ia} = \sum_{k \in K_a} V_{i,k,t0,t1}$.

7 INITIAL RESULTS AND CONCLUSIONS

ASC-U has been shown in practice to yield quality solutions and perform consistently. This consistent performance is characterized by increases in UAV performance resulting in corresponding increases in mission coverage and overall mission value. Furthermore, a recent thesis by Nannini (2006) has also shown ASC-U to perform consistently.

This paper demonstrates the implementation of approximate dynamic programming techniques in a discrete event simulation where simulation and optimization come together doing what they do best to solve a very complex scheduling problem. The approach takes an operational perspective so that solutions translate well to implementation in the real world or to a scheduling approach within a larger simulation.

Simulations allow measures of performance to be easily obtained from the dynamics and current state of the simulation. This capability allows for the following measures of performance to be easily obtained in ASC-U:

- Sensor Package Utilization,
- Percent Mission coverage by Mission type,
- Percent Mission coverage by Mission and UAV type,
- UAV utilization over time, and
- Ground Control Station load over time.

ASC-U also accounts for the effectiveness of remote viewing terminals (RVT) which allow users to see sensor data without any control of the actual UAV. Even though these RVTs do not effect the UAV schedule solution, they are easily inserted within the simulation tool.

The ASC-U model uses concepts from both simulation and optimization. Rather than using traditional simulation or optimization techniques in isolation, it utilizes each methodology to its advantage. Simulation is used to capture dynamic transitions of a sequential UAV scheduling decision problem, and optimization is used to allocate UAVs to mission areas within the scenario. ASC-U thus exploits the strengths of simulation and optimization to construct

good, timely solutions that neither optimization or simulation could achieve alone.

ACKNOWLEDGMENTS

The authors wish to thank United States Army TRADOC for support of the development of ASC-U.

REFERENCES

Ahner, D., L. Jackson, and D. Phillips. 2005. DAFS: A Low Resolution Modeling Approach: Architecture and Implementation. *Proceedings of The 10th Annual International Conference on Industrial Engineering Theory, Applications & Practice*. December 2005.

Buss, A. H., D. Ahner, and J. L. Ruck. 2006. Dynamic Allocation of Fires and Sensors (DAFS): A Low-Resolution Simulation For Rapid Modeling. In *Proceedings of the 2006 Winter Simulation Conference*. L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Buss, A. H. and P. J. Sanchez. 2005. Simple movement and sensing in discrete event simulation. In *Proceedings of the 2005 Winter Simulation Conference*. M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Havens, M. E. 2002. Dynamic Allocation of Fires and Sensors. Masters Thesis. Operations Research Department. Naval Postgraduate School, Monterey, CA.

Jackson, J. and D. Phillips. 2005. Using A Low Resolution Entity Level Modeling Approach. *The Bulletin of Military Operations Research: Phalanx*, 38-2: 15-26.

Nannini, C. 2006. Analysis of the Assignment Scheduling Capability for Unmanned Aerila Vehicles (ASC-U) Simulation Tool. Masters Thesis. Operations Research Department. Naval Postgraduate School, Monterey, CA

Lp_Solve. 2005. Lp_Solve Project Homepage. <http://sourceforge.net/projects/lpsolve/> [Accessed October 28, 2005].

Schruben, L. 1983. Simulation Modeling with Event Graphs. *Communications of the ACM*. 26(11): 957-963.

AUTHOR BIOGRAPHIES

DARRYL AHNER is an analyst at the U.S. Army Training and Doctrine Command analysis Center-Monterey. A Major in the United States Army, he received his Ph.D. in Systems Engineering (Operations Research) from Boston University. His e-mail address is dkahner@nps.edu.

ARNOLD BUSS is a Research Assistant Professor in the MOVES Institute at the Naval Postgraduate School. His e-mail address is <abuss@nps.edu>.

JOHN RUCK is a programmer for Rolands and Associates. A retired Naval officer, he received his BSE from Tulane University, and MS degrees from the University of Michigan and the Naval Postgraduate School. His e-mail address is <jlruck@nps.edu>.